

Python: module `esg.esg`

`esg.esg`

[index](#)

ESG markup support. See <http://www.earthsystemgrid.org>

Modules

[cdtime](#)

[re](#)

[string](#)

Classes

[Contact](#)

[ESGNode](#)

[Activity](#)

[Campaign](#)

[Ensemble](#)

[Investigation](#)

[Analysis](#)

[Experiment](#)

[Observation](#)

[Simulation](#)

[Project](#)

[Dataset](#)

[File](#)

[Institution](#)

[Metadata](#)

[Parameter](#)

[ParameterList](#)

[Person](#)

[Service](#)

[Format](#)

[Mapping](#)

[ObjRef](#)

[Participant](#)

[Qualified](#)

[Date](#)

[SimulationInput](#)

[Reference](#)

[SpaceRegion](#)

[TimeRegion](#)

[exceptions.Exception](#)

[ESGError](#)

class [Activity](#)([ESGNode](#))

Methods defined here:

__init__(self, id, name=None, description=None, rights=None, isPartOf=None, isDerivedFrom=None)
addDate(self, date)
addDerived(self, derived)
addNote(self, note)
addParticipant(self, participant)
addReference(self, reference)
write(self, fd)

Methods inherited from [ESGNode](#):

endwrite(self, fd)
startwrite(self, fd, nl=1, noname=0)

class [Analysis](#)([Investigation](#))

Method resolution order:

[Analysis](#)
[Investigation](#)
[Activity](#)
[ESGNode](#)

Methods defined here:

__init__(self, id, name=None, description=None, rights=None, parent=None)
write(self, fd)

Methods inherited from [Activity](#):

addDate(self, date)
addDerived(self, derived)
addNote(self, note)
addParticipant(self, participant)
addReference(self, reference)

Methods inherited from [ESGNode](#):

endwrite(self, fd)

startwrite(self, fd, nl=1, noname=0)

class ***Campaign***([Activity](#))

Method resolution order:

[Campaign](#)

[Activity](#)

[ESGNode](#)

Methods defined here:

__init__(self, id, name=None, description=None, rights=None)

write(self, fd)

Methods inherited from [Activity](#):

addDate(self, date)

addDerived(self, derived)

addNote(self, note)

addParticipant(self, participant)

addReference(self, reference)

Methods inherited from [ESGNode](#):

endwrite(self, fd)

startwrite(self, fd, nl=1, noname=0)

class ***Contact***

Methods defined here:

__init__(self, street=None, city=None, state=None, region=None, province=None, postcode=None, fax=None, url=None)

write(self, fd)

class ***Dataset***(ESGNode)

Methods defined here:

__init__(self, id, generatedBy=None, ofType=None, isPartOf=None, convention=None, timeCoverage=None, name=None)

Create a Dataset.

id: Unique string identifier

generatedBy: Investigation that generated the dataset.

ofType: String distinguishing characteristic within an investigation.

isPartOf: Dataset that contains this dataset, if any.

convention: String metadata convention ID, e.g. "CF-1.0"

timeCoverage: instance of TimeRegion

spaceCoverage: instance of SpaceRegion

addDate(self, date)

Set an event date.

ofType: String type of event, e.g., "start" or "stop".

datetime: String of form "yyyy-mm-dd hh:mi:ss"

addParameterList(self, qlist)

Add a parameter list.

qlist: idref of a parameter list.

setSpaceCoverage(self, sc)

setTimeCoverage(self, tc)

write(self, fd)

Methods inherited from ESGNode:

endwrite(self, fd)

startwrite(self, fd, nl=1, noname=0)

class ***Date***(Qualified)

Methods defined here:

__init__(self, type, encoding=None)

write(self, fd)

Methods inherited from Qualified:

setContent(self, content)

class ***ESGError***(exceptions.Exception)

Methods defined here:

__init__(self, args='Unspecified error from package esg')

__str__(self)

Methods inherited from exceptions.Exception:

__getitem__(...)

class ***ESGNode***

Abstract ESG node.

Methods defined here:

__init__(self, id, name=None)

endwrite(self, fd)

startwrite(self, fd, nl=1, noname=0)

write(self, fd)

Output to a file. fd is an open file descriptor.

class ***Ensemble***(Activity)

Method resolution order:

[Ensemble](#)

[Activity](#)

[ESGNode](#)

Methods defined here:

__init__(self, id, name=None, description=None, rights=None)

write(self, fd)

Methods inherited from Activity:

addDate(self, date)

addDerived(self, derived)

addNote(self, note)

addParticipant(self, participant)

addReference(self, reference)

Methods inherited from [ESGNode](#):

endwrite(self, fd)

startwrite(self, fd, nl=1, noname=0)

class [*Experiment*\(Investigation\)](#)

Method resolution order:

[Experiment](#)
[Investigation](#)
[Activity](#)
[ESGNode](#)

Methods defined here:

__init__(self, id, name=None, description=None, rights=None, parent=None)

write(self, fd)

Methods inherited from [Activity](#):

addDate(self, date)

addDerived(self, derived)

addNote(self, note)

addParticipant(self, participant)

addReference(self, reference)

Methods inherited from [ESGNode](#):

endwrite(self, fd)

startwrite(self, fd, nl=1, noname=0)

class [*File*\(ESGNode\)](#)

Methods defined here:

__init__(self, id, isPartOf=None, name=None, size=None)

write(self, fd)

Methods inherited from [ESGNode](#):

endwrite(self, fd)

startwrite(self, fd, nl=1, noname=0)

class ***Format***

Methods defined here:

__init__(self, uri, ofType)

write(self, fd)

class ***Institution***([ESGNode](#))

Methods defined here:

__init__(self, id, name=None, ofType=None, contact=None)

setContact(self, c)

write(self, fd)

Methods inherited from [ESGNode](#):

endwrite(self, fd)

startwrite(self, fd, nl=1, noname=0)

class ***Investigation***([Activity](#))

Method resolution order:

[Investigation](#)

[Activity](#)

[ESGNode](#)

Methods defined here:

__init__(self, id, name=None, description=None, rights=None)

Methods inherited from [Activity](#):

addDate(self, date)

addDerived(self, derived)

`addNote`(self, note)
`addParticipant`(self, participant)
`addReference`(self, reference)
`write`(self, fd)

Methods inherited from [ESGNode](#):

`endwrite`(self, fd)
`startwrite`(self, fd, nl=1, noname=0)

class ***Mapping***

Methods defined here:

`__init__`(self, authority, standardName=None)
`write`(self, fd)

class ***Metadata***([ESGNode](#))

Root node.

Methods defined here:

`__init__`(self)

`merge`(self, targetdsetid, sourcemd, sourcedsetid, newlistid, resultid=None, include=None, exclude=None, mustResolve='no')

Merge dataset sourcedsetid in container sourcemd, into dataset targetdsetid.

targetdsetid: string ID of target dataset, in self
sourcemd : metadata node containing source dataset
sourcedsetid: string ID of source dataset
newlistid : string ID for the new parameter list generated, with this ID, extra parameters are merged.
resultid : string ID of result dataset. Defaults to targetdsetid.
include : list of parameter names. If specified, only merge variables in the list.
exclude : list of parameter names. Do not merge variables in the list.
extralists : list of extra parameters lists
targetParams: list of parameters referenced by the source dataset
mustResolve : ["yes" | "no"] If yes, an error is raised whenever a parameter is not found in the target dataset. Default is "no".

Note: the merge is done in-place. If an exception occurs, the target dataset is restored to its previous state.

`setActivity`(self, activity)

`setDataset(self, dataset)`
`setFile(self, fileobj)`
`setInstitution(self, institution)`
`setParameter(self, parameter)`
`setParameterList(self, parameterList)`
`setPerson(self, person)`
`setService(self, service)`
`write(self, fd)`

Methods inherited from [ESGNode](#):

`endwrite(self, fd)`
`startwrite(self, fd, nl=1, noname=0)`

class *ObjRef*

Methods defined here:

`__cmp__(self, other)`
`__init__(self, idref)`
`write(self, fd, tag)`
 Write an object reference type (has an attribute 'idref')

class *Observation*([Investigation](#))

Method resolution order:

[Observation](#)
[Investigation](#)
[Activity](#)
[ESGNode](#)

Methods defined here:

`__init__(self, id, name=None, description=None, rights=None, parent=None)`
`write(self, fd)`

Methods inherited from [Activity](#):

addDate(self, date)
addDerived(self, derived)
addNote(self, note)
addParticipant(self, participant)
addReference(self, reference)

Methods inherited from [ESGNode](#):

endwrite(self, fd)
startwrite(self, fd, nl=1, noname=0)

class ***Parameter***([ESGNode](#))

Methods defined here:

__init__(self, id, name, description=None, activityRef=None, standardName=None, standardAutho
getMapping(self)
setMapping(self, mapping)
write(self, fd)

Methods inherited from [ESGNode](#):

endwrite(self, fd)
startwrite(self, fd, nl=1, noname=0)

class ***ParameterList***([ESGNode](#))

Methods defined here:

__init__(self, id, description=None, paramrefs=None, activityRef=None)
addParamRef(self, paramref)
write(self, fd)

Methods inherited from [ESGNode](#):

endwrite(self, fd)
startwrite(self, fd, nl=1, noname=0)

class ***Participant***(ObjRef)

Methods defined here:

__init__(self, idref, role)

write(self, fd)

 Write a participant type (has a role and idref) .

Methods inherited from ObjRef:

__cmp__(self, other)

class ***Person***(ESGNode)

Methods defined here:

__init__(self, id, firstName=None, lastName=None, contact=None, worksFor=None)

setContact(self, contact)

write(self, fd)

Methods inherited from ESGNode:

endwrite(self, fd)

startwrite(self, fd, nl=1, noname=0)

class ***Project***(Activity)

Method resolution order:

Project
Activity
ESGNode

Methods defined here:

__init__(self, id, name=None, description=None, rights=None, funding=None)

setTopic(self, topic)

write(self, fd)

Methods inherited from Activity:

addDate(self, date)

addDerived(self, derived)
addNote(self, note)
addParticipant(self, participant)
addReference(self, reference)

Methods inherited from [ESGNode](#):

endwrite(self, fd)
startwrite(self, fd, nl=1, noname=0)

class ***Qualified***

Methods defined here:

__init__(self, qual, tag=None)
setContent(self, content)
write(self, fd, tag)
 Write a qualified type (has an attribute 'type')

class ***Reference***

Methods defined here:

__init__(self, uri, reference=None)
setContent(self, content)
write(self, fd)
 Write a reference type (has a uri).

class ***Service*([ESGNode](#))**

Methods defined here:

__init__(self, id, name=None, description=None)

Methods inherited from [ESGNode](#):

endwrite(self, fd)
startwrite(self, fd, nl=1, noname=0)
write(self, fd)

Output to a file. fd is an open file descriptor.

class Simulation(Investigation)

Method resolution order:

Simulation
Investigation
Activity
ESGNode

Methods defined here:

__init__(self, id, name=None, description=None, rights=None)
setHardware(self, hardware)
setInput(self, input)
write(self, fd)

Methods inherited from Activity:

addDate(self, date)
addDerived(self, derived)
addNote(self, note)
addParticipant(self, participant)
addReference(self, reference)

Methods inherited from ESGNode:

endwrite(self, fd)
startwrite(self, fd, nl=1, noname=0)

class SimulationInput(Qualified)

Methods defined here:

__init__(self, ofType, input=None)

Methods inherited from Qualified:

setContent(self, content)
write(self, fd, tag)

Write a qualified type (has an attribute 'type')

class *SpaceRegion*

Methods defined here:

```
__init__(self, name=None, xrange=None, yrange=None, zrange=None)  
merge(self, sr)  
write(self, fd)
```

class *TimeRegion*

Methods defined here:

```
__init__(self, name=None, start=None, stop=None, encoding=None, calendar='gregorian')  
merge(self, tr)  
write(self, fd)
```

Functions

encodeIsoTime(timeTuple, toEncoding=None)

Encode a time tuple as an ISO 8601 time (see *parseIsoTime*).

timeTuple: tuple (year, month, day, hour, minute, second, timezone)

encodeTime(timeTuple, toEncoding=None)

Encode a time tuple.

timeTuple: tuple (year, month, day, hour, minute, second)

toEncoding: encoding string, such as 'yyyy-mm-dd hh:mi:ss'. If omitted
the string is generated based on the non-null values of timeTuple

mapIllegalToEntity(matchobj)

```
# Map reserved XML characters to entity references:  
# '<' --> &lt;  
# '>' --> &gt;  
# '&' --> &amp;  
# '\"' --> &quot;  
# '\"' --> &apos;  
# all other illegal characters are removed #"
```

mergeRange(ra, rb)

parseIsoTime(timeString)

Parse an encoded time string in ISO 8601 format, conforming to the

A tuple of the form (year, month, day, hour, minute, second, timezone)

In short, the encoding is:

```
"YYYY(-MM(-DD(Thh(:mi(:ss.s)?)?TZD)?)?)??"
```

(e.g. 1997-07-16T19:20:30.45+01:00)

where:

YYYY = four-digit year

MM = two-digit month (01=January, etc.)

DD = two-digit day of month (01 through 31)

hh = two digits of hour (00 through 23) (am/pm NOT allowed)

mi = two digits of minute (00 through 59)

ss = two digits of second (00 through 59)

s = one or more digits representing a decimal fraction of a second

TZD = time zone designator (Z or +hh:mm or -hh:mm)

parseTime(timeString, fromEncoding=r'yyyy(-mm(-dd(\s+hh(:mi(:ss)?)?))?)?")?

Parse an encoded time string, returning a tuple of the form
(year, month, day, hour, minute, second).

timeString: string representing an absolute time. If None, return
fromEncoding: input time encoding, a regular expression (see the re module)

defaults to "yyyy(-mm(-dd(\s+hh(:mi(:ss)?))?)?)?")?"

An exception is raised if the time string and encoding don't match.

writes(fd, tag, s)

Write a string element.

Data

```
defaultTimeEncoding = r'yyyy(-mm(-dd(\s+hh(:mi(:ss)?))?)?)?")?
```

```
namespace = 'http://www.earthsystemgrid.org/'
```

```
nsprefix = 'esg'
```

```
schemaLocation = 'http://www.ucar.edu/schemas/esg.xsd'
```